CLAIMS

What is claimed is:

1. A squaring circuit comprising:

a host processor to compute the square of a long integer value by recursively reducing the square of said long integer value to a combination of the squares of reduced length integer values of a predetermined length, wherein at each step of said recursion said host processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer values;

a co-processor connected to said host processor to compute the squares of said reduced length integer values by further recursively reducing the squares of said reduced integer values into a combination of squares of hardware-length integer values that can be squared by hardware logic circuits, wherein at each step of said recursion said co-processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer values; and

one or more hardware logic circuits to square said hardware-length integer values.

2. The squaring circuit of claim 1 wherein said one or more hardware logic circuits comprise a separate hardware logic circuit for each hardware-length integer value to be squared.

3. The squaring circuit of claim 1 wherein said host processor further computes the product of two long integer values by computing the difference between the square of the sum of said two long integer values and the square of the difference of said two long integer values.

4. The squaring of claim 3 wherein said host processor divides said sum of said two long integer values and said difference of said two long integer values by two before computing said squares of said sum and said difference.

5. The squaring circuit of claim 4 wherein said host processor adds the smaller value of said two long integer values to the difference of said squares to form said final product.

6. The squaring circuit of claim 1 wherein said host processor further computes a power of a long integer value by computing successive squares of said long integer value and by computing the product of selected ones of said successive squares corresponding to binary "1"s in said power.

7. The squaring circuit of claim 6 wherein said host processor computes said product of selected ones of said successive squares by computing the difference between the squares of the sum and the difference of said successive squares.

8. The squaring circuit of claim 1 further comprising randomly ordering each set of three ending integer values in at least one stage of said recursion.

9. A method of squaring a long integer value comprising:

recursively reducing the square of said long integer value in a host processor to a combination of squares of reduced integer values, wherein at each step of said recursion said host processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer values;

computing the squares of said reduced-length integer values in a co-processor connected to said host processor by further recursively reducing the squares of said reduced integer values into a combination of squares of hardware-length integer values that can be squared by hardware logic circuits, wherein at each step of said recursion said co-processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer value; and

computing the squares of said hardware-length integer values in one or more hardware logic circuits.

10. The method of claim 9 wherein computing said square of said hardware length integer values comprises computing said squares of said hardware-length values in separate hardware logic circuits.

11. The method of claim 9 further comprising computing the product of two long integer values in said host processor by computing the difference between the squares of the sum and the difference of said two long integer values.

12. The method of claim 11 wherein computing the difference between said squares of said sum and said difference of said two long integer values comprises dividing said sum and said difference of said two long integer values by two before computing said squares of said sum and said difference.

13. The method of claim 12 wherein computing said difference between said squares of said sum and said difference of said two long integer values further comprises adding the smaller of said two long integer values to said difference of said squares of said sum and said difference to form said final product.

14. The method of claim 9 further comprising computing a power of said long integer value by computing successive squares of said long integer value and by computing the product of selected ones of said successive squares corresponding to binary "1"s in said power.

15. The method of claim 14 wherein computing the product of selected ones of said successive squares corresponding to binary "1"s in said power comprises computing the difference between the squares of the sum and the difference of said successive squares.

16. The method of claim 9 further comprising randomly ordering each set of three ending integer values in at least one stage of said recursion.

17. A squaring circuit comprising:

a host processor to compute the square of a long integer value by recursively reducing said square of said long integer value into a combination of squares of reduced integer values, wherein at each step of said recursion said host processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer values, and wherein at each step of said recursion said host processor randomly orders said ending integer values; and

a co-processor connected to said host processor to compute the squares of said reduced-length integer values.

18. The method of claim 17 further comprises one or more hardware logic circuits to square hardware-length integer values and wherein said co-processor computes the squares of said reduced-length integer values by further recursively reducing the squares of said reduced length integer values into a combination of squares of hardware length integer values whose values are computed by said hardware logic circuits.

19. The squaring circuit of claim 17 further comprising a noise generator to generate a random sequence used for randomly ordering said ending integer values by combining squares of further-reduced-length integer values performed by a hardware logic circuit.

20. The squaring circuit of claim 18 wherein said one or more hardware logic circuits comprise a separate hardware logic circuit for each further-reduced-length integer value to be squared.

21. The squaring circuit of claim 17 wherein said host processor further computes the product of two long integer values by computing the difference between the squares of the sum and the difference of said two long integer values.

22. The squaring of claim 21 wherein said host processor divides said sum and said difference by two before computing the squares of said sum and said difference.

23. The squaring circuit of claim 22 wherein said host processor adds the smaller value of said two long integer values to the difference of said squares to form said final product.

24. The squaring circuit of claim 17 wherein said host processor further computes a power of a long integer value by computing successive squares of said long integer value and by computing the product of selected ones of said successive squares corresponding to binary "1"s in said power.

25. The squaring circuit of claim 24 wherein said host processor computes said product of selected ones of said successive squares by computing the difference between the squares of the sum and the difference of said successive squares.

26. The squaring circuit of claim 17 further comprising randomly ordering each set of three ending integer values in at least one stage of said recursion.

27. A circuit for multiplying two long integer values including secret data while hiding the value of the secret data, said circuit comprising:

a host processor to compute the product of said two long integer values by recursively reducing a product of said two long integer values to a combination of products of reduced length integer values of a predetermined length, wherein at each step of said recursion said host processor expresses starting integer values as a combination of products of three ending integer values of one-half the length of said starting integer values, and wherein at each step of said recursion said host processor randomly orders said ending integer values; and

a co-processor connected to said host processor to compute products of said reduced-length integer values.

28. The circuit of claim 27 further comprising a noise generator to generate a random sequence used to randomly order said ending integer values.

29. A method of squaring a secret long integer value using an insecure co-processor connected to a secure host processor, said method comprising:

recursively reducing a square of said long integer value to a combination of squares of reduced-length integer values of a predetermined length in said secure host processor, wherein at each step of said recursion said secure host processor reduces starting integer values to a combination of squares of three ending integer values of one-half the length of said starting integer values;

randomly ordering said three ending integer values at each stage of said recursion; and

computing squares of said reduced-length integer values in said insecure processor and returning the result to said host processor to compute the final square of said long integer value.

30. The method of claim 29 where randomly ordering said ending integer values at each stage of said recursion comprises generating a random sequence in a noise generator to control ordering of said ending integer values.

31. A method of multiplying two long integer values including secret data while hiding the value of said secret data, said method comprising:

computing the product of said two long integer values in a secure host processor by

recursively reducing the product of said long integer values to a combination of products of reduced-length integer values, wherein at each step of said recursion said host processor expresses the product of starting integer values as a combination of the products of three ending integer values of one-half the length of said starting integer values;

at each stage of said recursion, randomly ordering said three ending integer values; and

computing the product of each set of final ending integer values output from said host

processor and returning said products to said host processor to use in computing the final product of said two long integer values.

32. The method of claim 31 wherein randomly ordering said ending integer values comprises generating a random sequence in a noise generator to control ordering of said ending integer values.

33. A method of designing a logic circuit to be manufactured comprising:

defining an indexing parameter;

for values of said indexing parameter extending from a desired value to a minimum value, defining a recursive logic circuit with said indexing parameter set to a current value as interconnections between pre-defined logic circuits and one or more instances of said recursive logic circuit with said indexing parameter less than said current value;

defining a base logic circuit with an indexing parameter equal to said minimum value as interconnections between pre-defined logic circuits; and

processing said definitions of said recursive logic circuits and said base logic circuit for said desired value of said indexing parameter to produce a definition of said recursive logic circuit for said desired value of said indexing parameter in terms of said pre-defined logic circuits.

34. The method of claim 33 wherein defining an indexing parameter comprises defining said indexing parameter as the wordlength of a data value processed by said logic circuit.

35. The method of claim 33 wherein defining a recursive logic circuit with said indexing parameter set to a current value as interconnections between predefined logic circuits and one or more instances of said logic circuit with said indexing parameter less than said current value comprises defining a recursive logic circuit as interconnections between predefined logic circuits and one ore more instances of said recursive logic circuit with said indexing parameter equal to said current value less one.

36. The method of claim 33 wherein defining a recursive logic circuit with said indexing parameter set to a current value as interconnections between predefined logic circuits and one ore more instances of said recursive logic circuit with said indexing parameter less than said current value comprises defining said recursive logic circuit as interconnections between predefined logic circuits and an instance of said logic circuit with said indexing parameter equal to half said current value.

37. The method of claim 33 wherein defining a recursive logic circuit with said indexing parameter set to a current value as interconnections between predefined logic circuits and one ore more instances of said recursive logic circuit with said indexing parameter less than said current value comprises describing said recursive logic circuit using VHDL.

38. The method of claim 37 wherein processing said definitions of said recursive logic circuits and said base logic circuit for said desired value of said indexing parameter to produce a definition of said recursive logic circuit for said desired value of said indexing in terms of said predefined logic circuits comprises processing said definitions using a recursive VHDL compiler.

39. The method of claim 37 wherein processing said definitions using a recursive VHDL compiler comprises:

preprocessing recursive VHDL definitions to produce a modified, non-recursive VHDL description of said final logic circuit; and

processing said non-recursive VHDL definition of said final logic circuit using a non-recursive VHDL compiler to produce a definition of said final logic circuit in terms of interconnections between said predefined logic circuits.

40. The method of claim 33 in which said final logic circuit is a circuit for multiplying long integers.

41. The method of claim 33 in which said logic circuit is a circuit for squaring long integers.

42. The method of claim 33 wherein defining a base logic circuit with said indexing parameter equal to said minimum value as interconnections between predefined logic circuits comprises defining a base logic circuit with an indexing parameter equal to two.